

Very short answer questions. "True" and "False" are considered very short answers.

(2) Clock rates have grown by a factor of 1000 while power consumed has only grown by a factor of 30. How was this accomplished?

(1) The use of large, multilevel caches can substantially reduce the memory bandwidth demands of a processor.

(1) As minimum feature sizes decrease, what happens to wire resistance?

(2) Does reducing the minimum feature size affect power density? How?

(1) It is not necessary to simulate very many instructions in order to get an accurate performance measure of the memory hierarchy.

(1) A program's locality behavior stays constant over the run of an entire program.

(1) The instruction set architecture has a significant impact on the ability to pipeline a processor.

(2) What does UMA stand for? Which is more sensitive to where data is placed, an UMA machine or a NUMA machine?

(1) Does peak performance track observed performance?

(1) Which is on average more effective, dynamic or static branch prediction?

(1) Predicting the direction of a branch is not enough. What else is necessary?

(1) Do benchmarks remain valid indefinitely?

(2) What is Amdahl's law (in words)?

(3) Write down the 3-term CPU performance equation developed in class.

(1) The instruction set architecture has little impact on the implementability of a virtual machine monitor.

(1) Write down the average memory access time equation.

(2) Why are there multiple dies per silicon wafer? Why not just fabricate one huge die per wafer?

(2) What is the relationship between speculation and power consumption?

(2) Prefetching is one technique for reducing the Miss Rate. List 2 others.

(2) Lowering the associativity is one technique for reducing the Hit Time. List 2 others.

(2) Increasing the block size is one technique for reducing the cache power consumption. List 2 others.

(3) What is the goal of the memory hierarchy? What two principles make it work? (Two types of the same principle, actually).

- (2) What is the difference between coherence and consistency?
- (2) How is communication between processors done in a shared memory machine?
- (1) Which type of cache miss can be changed by altering the mapping scheme?
- (1) Which type of cache miss can be reduced by using longer lines?
- (1) Which type of cache miss can be increased by using longer lines?
- (1) What hardware structure is necessary in order to make a snooping protocol work?
- (2) What is the "threshold voltage"?
- (1) What pipeline hazard can be avoided by "throwing money at the problem"?
- (1) What pipeline hazard can be avoided using a technique known as value prediction?
- (2) Why do CMOS processors avoid accessing items that are located off-chip?
- (2) How many entries are there in a (6,3) Gshare branch predictor? How many bits?
- (2) What are the two instructions used in RISC machines to support atomicity?

(2) What is the main difference between a commodity cluster and a custom cluster?

(2) Relaxing the requirement that Writes complete before Reads yields a model known as:

(2) How is communication between processors done in a message passing machine?

(4) What are the two main ways to define performance? How do they differ? Give an example task for each.

Short Answers:

(3) Why is it difficult to come up with good benchmarks for parallel processors?

(4) Which is more expensive to build - a shared memory machine, or a message passing machine? Why?

(4) What are the 4 types of data hazards? Which one is not really a hazard? Why not?

(3) In an MOS device, there is a gate, drain, and source. Briefly explain how this device works. Pictures are welcome.

(4) What are the two biggest challenges in parallel processing? In other words, what two things are keeping parallel processors from being the dominant architecture?

(4) What is the primary difference between Scoreboarding and Tomasulo's algorithm? What hardware feature makes Tomasulo's work?

(6) Understanding the hardware can influence how you write programs. Give at least 2 examples of how you might write software differently for a heavily pipelined machine versus a non-pipelined one.

(4) What is the definition of a basic block? Why is there a desire to create larger ones?

(4) What is a benchmark program? What is the **perfect** benchmark?

(4) What does SIMD stand for? What does MIMD stand for? Give 1 advantage and 1 disadvantage to using a MIMD processor.

(4) Supporting precise interrupts in machines that allow out of order completion is a challenge. Briefly explain why, and give two different techniques that can be used to provide precise interrupts.

(4) Why is branch prediction important? List 3 existing dynamic Branch Prediction strategies in order of (average) decreasing effectiveness.

(5) Compare and contrast Superscalar and VLIW. Describe each, and list the advantages and disadvantages of each approach.

(5) Snooping is one approach to providing coherence - state what the other main approach is, and briefly outline how each of them work.

(3) Assuming a 16-bit address and a 256-byte Direct Mapped cache with a linesize=4, show how an address is partitioned/interpreted by the cache.

(3) Assuming a 16-bit address and a 160-byte 5-way SA cache with a linesize=2, show how an address is partitioned/interpreted by the cache.

(3) Given a 1 Megabyte physical memory, a 24 bit Virtual address, and a page size of 2K bytes, write down the number of entries in the Page Table, and the width of each entry.

(4) Given a 1 Megabyte physical memory, a 32 bit Virtual address, and a page size of 2K bytes, write down the number of entries in the Page Table, and the width of each entry. Is there a problem with this configuration? If so, how can you fix it?

(4) There are at least two types of control flow changes that standard dynamic branch predictors have trouble with. Explain what the two are and why they are a problem, and give the technique used to successfully deal with one of them.

(4) The memory system presents challenges to ILP designers as well. What is it about the memory system that makes it hard for compilers to optimize code, and also for execution units to achieve maximal performance? (This is not a question about technology, it's a higher-level question).

(6) In class, we talked about the cycle by cycle steps that occur on different interrupts. For example, here is what happens if there is an illegal operand interrupt generated by instruction i+1:

	1	2	3	4	5	6	7	8	9	
10										
i	IF	ID	RR	EX	MEM	WB				
i+1		IF	ID	RR	EX	MEM	WB	<- Interrupt detected		
i+2			IF	ID	RR	EX	MEM	WB	<- Instruction Squashed	
i+3				IF	ID	RR	EX	MEM	WB <- Trap Handler fetched	
i+4					IF	ID	RR	EX	MEM	WB

Fill out the following table if instruction i+1 experiences a fault in the EX stage (arithmetic exception, for example):

	1	2	3	4	5	6	7	8	9	10	11
i	IF	ID	RR	EX	MEM	WB					
i+1		IF	ID	RR	EX	MEM	WB				
i+2			IF	ID	RR	EX	MEM	WB			
i+3				IF	ID	RR	EX	MEM	WB		
i+4					IF	ID	RR	EX	MEM	WB	
i+5						IF	ID	RR	EX	MEM	WB

What happens in this case?

	1	2	3	4	5	6	7	8	9	10	11
i	IF	ID	RR	EX	MEM	WB	<- Data write causes Page Fault				
i+1		IF	ID	RR	EX	MEM	WB	<- Illegal Opcode			
i+2			IF	ID	RR	EX	MEM	WB			
i+3				IF	ID	RR	EX	MEM	WB		
i+4					IF	ID	RR	EX	MEM	WB	
i+5						IF	ID	RR	EX	MEM	WB

(2) What is the maximum number of exceptions that could happen at one time in the above machine?